1 Short Answer

1. (5 pts) In the substitution cipher, each letter of the English alphabet is replaced by a corresponding output letter, with each input letter mapping to a unique output letter. Explain how you would use a frequency attack to decrypt a ciphertext encrypted using the substitution cipher.

Solution: To perform a frequency attack on ciphertext encrypted using the substitution cipher, find the most frequently occurring letter. This will most likely correspond to the letter 'e'. The second most frequently occurring letter will most likely correspond to the next most frequently occurring letter (the letter 't'). Proceeding in this way for the first few most frequently occurring letters, will get you some of the text—but might leave some ambiguities. It is then necessary to look at digrams (two letters) to further narrow down the choices. Digrams take advantage of the fact that certain pairs of letters occur more often than other pairs of letters (e.g. 'th' is more common than 'dz'). Once you have used the digrams, it should be possible to fill in many of the remaining letters.

2. (5 pts) Explain the mathematical meaning of the Euler Phi function \( \phi(n) \).

Solution: The Euler Phi function \( \phi(n) \) counts the amount of numbers \( a \) with \( 1 \leq a \leq n \) such that \( \gcd(a, n) = 1 \). That is, the amount of numbers relatively prime to \( n \), and less than \( n \).

3. (5 pts) Describe the Miller-Rabin algorithm for finding a random prime number.

Solution: The Miller-Rabin algorithm is based upon two principles: (1) That \( a^{n-1} \neq 1 \pmod{n} \) means \( n \) is not prime, and \( a^{n-1} = 1 \pmod{n} \) means \( n \) is most likely prime; (2) if \( x^2 = y^2 \pmod{n} \) then \( \gcd(x - y, n) \) is a factor of \( n \). Combining these two facts, the Miller-Rabin algorithm looks at various powers \( a^{2^j r} \pmod{n} \) to see whether they are \(-1\), in which case it declares it probably prime. If \( a^{2^j r} \pmod{n} = 1 \) but \( a^{2^{j-1} r} \pmod{n} \neq -1 \) then using reason (2) we can find a factor of \( n \) so \( n \) is not prime. Miller-Rabin proceeds until it reaches the \( k \)-th step where \( n - 1 = a^{2^k r} \) with \( r \) odd. Miller-Rabin can be repeated for different choices of \( a \).

4. (5 pts) Explain why using two prime numbers \( p \) and \( q \) that are close to each other in value is a bad idea for RSA.

Solution: One may simply use Fermat factorization to easily find \( p \) and \( q \). Once finding \( p \) and \( q \), one may find \( \phi(n) \) and hence can easily solve \( de = 1 \pmod{\phi(n)} \).

2 Calculations

1. (5 pts) Calculate \( 7^{-1} \pmod{11} \).

Solution: Observe that \( 7 \cdot 8 = 56 \), and \( 56 = 1 \pmod{11} \). Hence \( 7^{-1} \pmod{11} = 8 \).

2. (5 pts) Suppose Alice chooses \( n = 35 \) as her RSA modulus, and chooses \( e_A = 7 \) as her public exponent. Hence her public key is \( (n, e_A) \). Calculate her private decryption exponent \( d_A \).

Solution: Observe that \( n = 5 \cdot 7 \), and hence \( \phi(n) = (5-1)(7-1) = 24 \). The private decryption exponent \( d_A \) satisfies

\[ e_A d_A = 1 \pmod{24} \]

and hence \( d_A = 7 \). (So, encryption and decryption would be the same for this example!)
3. (10 pts) In a (3, 5) Shamir secret sharing scheme using $p = 17$, the following secrets were generated:

$$A : (2, 13)$$
$$B : (4, 5)$$
$$C : (6, 13).$$

Find the secret $M$. If user $E$ was given the secret $(10, *)$, find the missing value $*$. 

Solution: The best way to approach this problem is to use LaGrange Interpolation to find $M$ and to find the interpolating polynomial.

Calculating the LaGrange interpolating polynomial gives $s(x) = 3 + x + 2x^2 \pmod{17}$. Here is how:

$$s(x) = 13 \frac{x - 4}{2 - 4} \frac{x - 6}{2 - 6} + 5 \frac{x - 2}{4 - 2} \frac{x - 6}{4 - 6} + 13 \frac{x - 2}{6 - 2} \frac{x - 4}{6 - 4} \pmod{17}.$$ 

Reducing gives the desired result for $s(x)$. The secret can is simply the polynomial evaluated at 0. Hence $M = s(0) = 3$. The value for $*$ is the polynomial evaluated at $x = 10$, that is $* = s(10) = 9$.

3 Advanced Understanding

1. (10 pts) Suppose $E_K(M)$ is the DES encryption of a message $M$ using the key $K$. We showed in the homework that DES has the complementation property, namely that if $y = E_K(M)$ then $\overline{y} = E_{\overline{K}}(\overline{M})$, where $\overline{M}$ is the bit complement of $M$. That is, the bitwise complement of the key and the plaintext result in the bitwise complement of the DES ciphertext. Explain how an adversary can use this property in a brute force, chosen plaintext attack to reduce the expected number of encryptions that would be tried from $2^{55} \to 2^{54}$. (Hint: Consider a chosen plaintext set of $(M_1, C_1)$ and $(\overline{M_1}, C_2)$).

Solution: This is a tricky little problem with a deceptively simple looking answer.

Let $K$ be the key we wish to find. Use the hint. Then $C_1 = E_K(M_1)$ and $C_2 = E_{\overline{K}}(\overline{M_1})$. Now, suppose we start a brute force attack by encrypting $M_1$ with different keys. If, when we use $K_j$ we get $E_{K_j}(M_1) = C_1$ then we are done and the key we desire is $K = K_j$. However, when we use $K_j$ we can eliminate another key. Here is how. If $E_{K_j}(M_1) = \overline{C_2}$ then we know (by complementation property) that $E_{\overline{K_j}}(\overline{M_1}) = C_2$. Hence, if this happens, we know the key is $K_j$ since $K_j$ would decrypt $C_2$ to get $\overline{M_1}$. We are effectively testing two keys for the price of one! Hence, the key space is cut in half and we only have to search an average of $2^{54}$.

2. Let us consider the following DES-like encryption method. We start with a message $M$ of $2n$ bits, and divide it into two blocks of length $n$ bits, $M = [M_0, M_1]$. Our DES-like encryption algorithm will proceed in rounds, where the output of round $j$ is

$$M_{j+2} = M_j \oplus f(K_{j+1}, M_{j+1})$$

where $K_{j+1}$ is a subkey for round $j$ that is determined from $K$.

(a) (10 pts) Suppose that the decryptor receives $M_d$ and $M_{d-1}$, and that he knows the subkeys $K_j$, explain how to decrypt $M_d$ and $M_{d-1}$ in order to obtain $M = [M_0, M_1]$.

(b) (10 pts) Now suppose that $n = 3$, and that

$$f([x_1, x_2, x_3], [y_1, y_2, y_3]) = (x_1x_2y_1, x_2x_3y_2y_3, x_1x_3y_3).$$

We define $K = (k_1, k_2, \cdots, k_8)$ to be the eight bit key (hence $k_j$ is either 0 or 1). Suppose the subkeys are given by the key schedule:

$$K_1 = (k_1, k_2, k_3) \quad K_2 = (k_2, k_3, k_4)$$
$$K_3 = (k_3, k_4, k_5) \quad K_4 = (k_4, k_5, k_6).$$


3. We now look at a Feistel round. Given \( M_d, M_{d-1}, \) and \( K \) to get \( M_{d-2} \) use the fact that

\[
M_d = M_{d-2} \oplus f(K_{d-1}, M_{d-1}).
\]

Thus we now know \( M_d, M_{d-1}, \) and \( M_{d-2} \). We may use

\[
M_{d-1} = M_{d-3} \oplus f(K_{d-2}, M_{d-2})
\]

in a similar manner to get \( M_{d-3} \), and so on until we get \( M_1 \) and \( M_0 \).

(b) (Note: Due to a typing error on the original handout, this problem will be not be graded. Full credit will be given to all students.) The intended solution follows.

Observe that \( M_0 = [101] \) and \( M_1 = [010] \). The first key \( K_1 = (k_1, k_2, k_3) = [000] \). Plugging into \( f(K_1, M_1) \) we get

\[
M_2 = M_0 \oplus f(K_1, M_1) = 101 \oplus 000 = 101.
\]

Next \( M_3 = M_1 \oplus f(K_2, M_2) \). In this round \( K_2 = [010] \) and \( M_2 = [101] \) (which we just calculated). Then \( f(K_2, M_2) = [000] \) and

\[
M_3 = 010 + 000 = 010.
\]

Thus \([M_2, M_3] = 101010\).

3. We now look at a 3-person group encryption scheme based on the same principle as RSA. Suppose that some trusted entity generates two primes \( p \) and \( q \) and forms \( n = pq \). Now, instead of choosing \( e_A \) and \( d_A \) (as in RSA), the trusted entity chooses \( k_1, k_2, \) and \( k_3 \) such that \( gcd(k_j, n) = 1 \) and

\[
k_1k_2k_3 = 1 \pmod{\phi(n)}.
\]

The three users \( A, B, \) and \( C \) are given the following keys

\[
A : (k_1, k_2, n)
B : (k_2, k_3, n)
C : (k_1, k_3, n)
\]

(a) (5 pts) Suppose user \( A \) generates a message \( m \) such that \( gcd(m, n) = 1 \). \( A \) wants to encrypt \( m \) so that both \( B \) and \( C \) can decrypt the ciphertext. To accomplish this, \( A \) forms the ciphertext

\[
y = m^{k_1k_2} \pmod{n}.
\]

Explain how \( B \) would decrypt \( y \), and explain how \( C \) would decrypt \( y \).

Solution: \( B \) would simply do \( y^{k_3} \pmod{n} = m^{k_1k_2k_3} \pmod{n} \). By Euler’s Theorem \( m^{k_1k_2k_3} = m \pmod{n} \). \( C \) would do the same.

(b) (10 pts) Suppose \( A \) and \( B \) have been collaborating on some class project and have produced the message \( m \) (with \( gcd(m, n) = 1 \)). They would like to create a ciphertext that they can send \( C \) so that only \( C \) can decrypt it, and such that once encrypted neither \( A \) nor \( B \) alone can decrypt the ciphertext to recover \( m \). Explain how this can be accomplished.
Solution: Observe that $A$ and $B$ can encrypt their message successively:

$$m^{k_1k_2} \rightarrow (m^{k_1k_2})^{k_2k_3} = m^{k_2} \pmod{n}.$$

To decrypt, $C$ needs to raise $m^{k_2}$ to the $k_1k_3$ power:

$$m = (m^{k_2})^{k_1k_3} \pmod{n}.$$

Now, if $A$ or $B$ loses $m$, then they can’t recover $m$ since neither of them, by themselves, knows both $k_1$ and $k_3$.

(c) (15 pts) In some scenarios, it is necessary to have a digital document signed by multiple participants. For example, a check issued by a company might need to be signed by both the issuer and a representative in the financial department before the check is valid. The above idea can be used to accomplish this. Suppose that $n = pq$ for two primes $p$ and $q$, and that the trusted entity chooses $k_1$, $k_2$, and $k_3$ such that $gcd(k_j, n) = 1$ and $k_1k_2k_3 = 1 \pmod{\phi(n)}$. Suppose that a digital document $m$ (an integer such that $gcd(m, n) = 1$) needs to be signed by two individuals, the Issuer and the Financier, in order for the document to be valid. Further, $(k_1, n)$ is secretly given to the Issuer, $(k_2, n)$ is secretly given to the Financier, and $(k_3, n)$ is made public. Determine a procedure that will allow

- The Issuer to first sign the document $m$ to produce the signature $y_I$.
- The Financier to verify that the signed document $y_I$, which he receives from the Issuer, corresponds to $m$.
- The Financier to sign $y_I$ to produce $y_F$, which contains both the Issuer’s and the Financier’s signatures.
- The document $y_F$ to be verified by any other person as corresponding to $m$, and having been signed by both the Issuer and the Financier.

Solution:

- First, the Issuer must sign using his private information. By following a procedure like RSA-signatures, the Issuer creates the signed document

$$y_I = m^{k_1} \pmod{n}.$$

- The Financier can verify that $y_I$ in fact corresponds to $m$ by using his private information $(k_2, n)$ and the public information $(k_3, n)$:

$$m = (y_I)^{k_2k_3} \pmod{n}.$$

The key thing to note here is that the public information is available to everyone.

- The Financier creates $y_F = y_I^{k_2} \pmod{n} = m^{k_1k_2} \pmod{n}$.

- The document $y_F$ can be verified as corresponding to $m$ by using the public key $(k_3, n)$

$$m = y_F^{k_3} \pmod{n}.$$

Since this verifies, and since $k_1$ and $k_2$ are private, it must have been signed by both the Issuer and the Financier.

(Note: One may also have done $y_I = m^{k_1k_3} \pmod{n}$, in which case verification would have been $y_I^{k_2} \pmod{n}$.)